Assignment 1 Report

Cansu ASLAN
2210356079

Introduction

In this assignment, Convolutional Neural Networks (CNNs) were used to classify images from the Food-11 dataset. Two approaches were tested: training models from scratch and using transfer learning with a pre-trained network. Different training settings such as learning rate, batch size, and dropout were tried. Model performance was evaluated using validation accuracy, test accuracy, and confusion matrices.

Part 1

In the first part of the assignment, two custom CNN architectures were implemented and trained from scratch using the Food-11 dataset. The goal was to explore how CNNs perform without the use of pre-trained weights, and how architectural choices and hyperparameters affect training.

The first model, a basic CNN, consisted of five convolutional layers with batch normalization and ReLU activations. Max pooling was used after each block to reduce spatial resolution. The output feature maps were flattened and passed through two fully connected layers to produce the final class predictions.

The second model, ResidualCNN, was based on the same structure but included residual connections to help the network learn more effectively in deeper layers. These connections allowed gradients to flow more easily through the network and helped mitigate issues such as vanishing gradients. Each residual block contained two convolutional layers, and when input-output dimensions didn't match, a 1×1 convolution was applied in the skip path.

For training, different combinations of learning rates (0.001, 0.0005, 0.0001) and batch sizes (32, 64) were tested. All models were trained using the Adam optimizer and CrossEntropyLoss for 50 epochs. The best-performing model, with a learning rate of 0.0001 and batch size 32, achieved a validation accuracy of 66.55%.

To improve generalization, data augmentation was applied during training. Augmentations included random crops, horizontal flips, and small rotations. The validation and test sets were only resized and normalized to ensure consistent and noise-free evaluation.

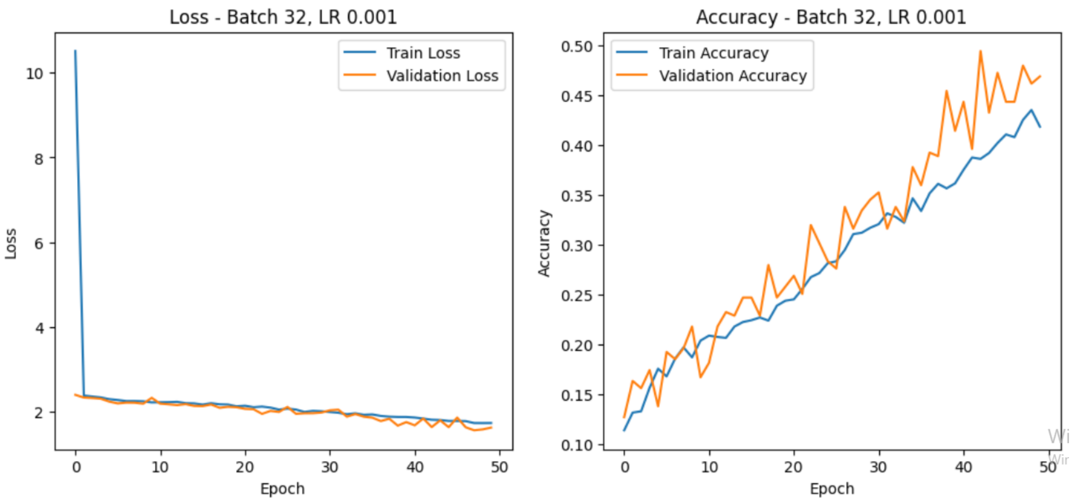
Later, dropout was added to the best model to test its effect on regularization. Two dropout rates were evaluated (0.3 and 0.5), both applied after the first fully connected layer. The 0.3 dropout slightly improved performance, while 0.5 resulted in mild underfitting, confirming that moderate regularization was beneficial.

For each batch size and learning rate combination, training loss, training accuracy, and validation accuracy were plotted over 50 epochs. These plots helped visualize model behavior, including convergence speed and signs of overfitting.

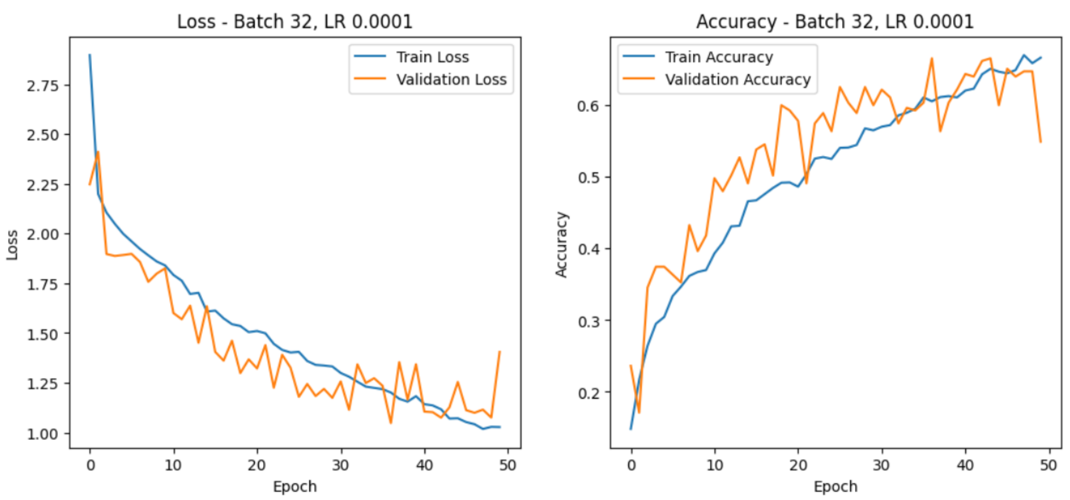
Models with lower learning rates generally showed more stable training, while higher rates sometimes led to early plateaus or fluctuations. The best performance was observed with batch size 32 and learning rate 0.0001, which showed a steady decrease in loss and consistent validation accuracy improvement.

Plots for all configurations are included below, along with their corresponding validation accuracy values. These results guided the selection of the best model for further experiments with dropout.

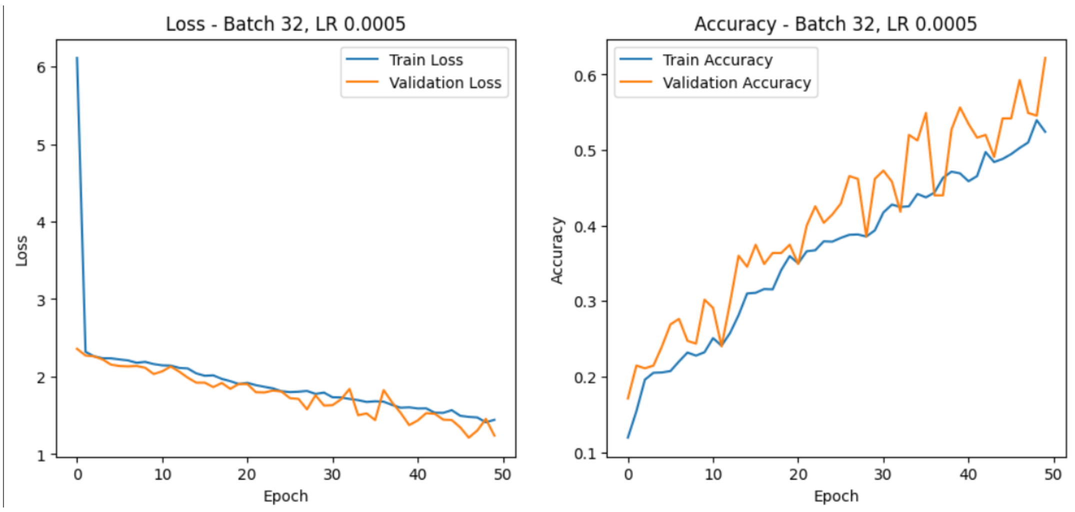
Basic CNN



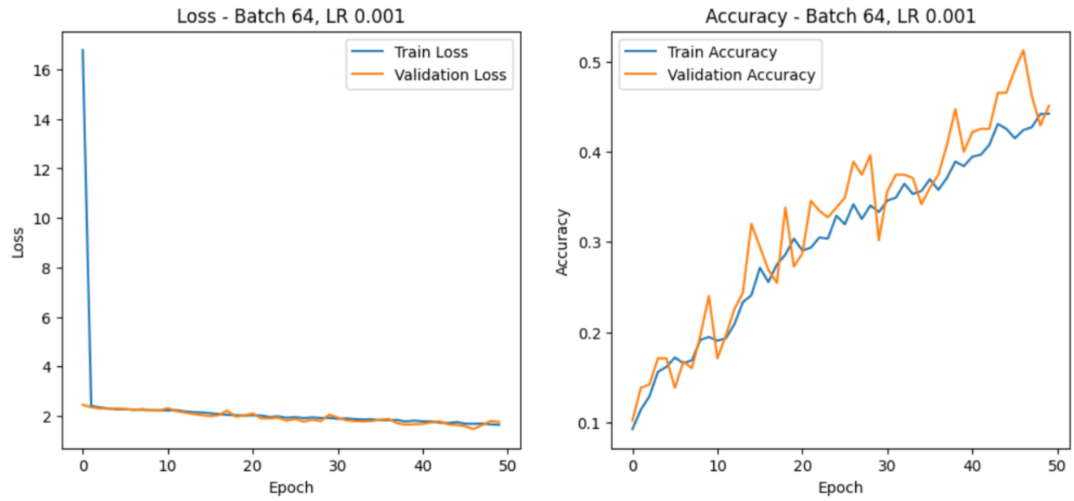
For Basic CNN Model, the values obtained at the end of 50 epochs with 32 batch size and 0.001 learning rate are: Train Loss: 1.7240, Train Acc: 0.4186, Val Loss: 1.6160, Val Acc: 0.4691



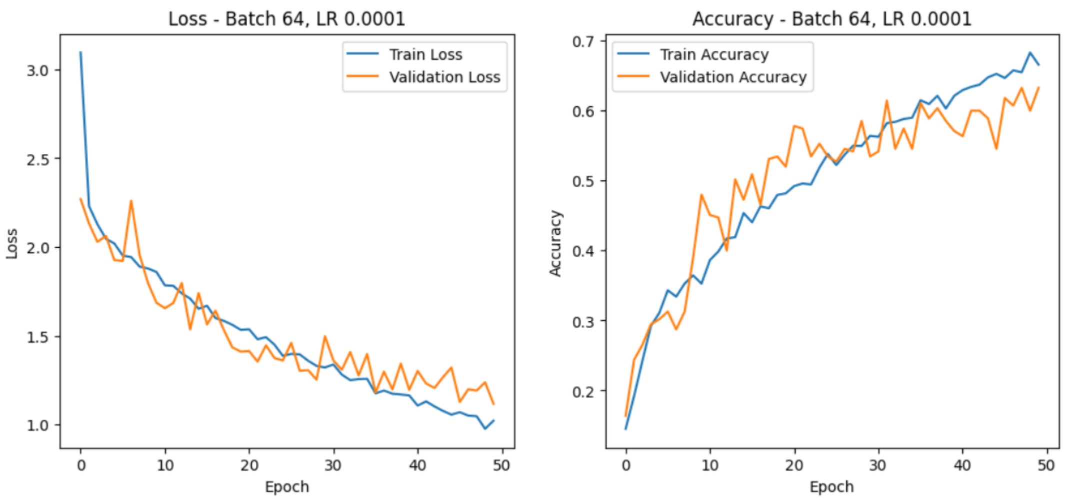
For Basic CNN Model, the values obtained at the end of 50 epochs with 32 batch size and 0.0001 learning rate are: Train Loss: 1.1830, Train Acc: 0.6109, Val Loss: 1.3432, Val Acc: 0.6218 These are the values with the best validation accuracy among the basic CNN models.



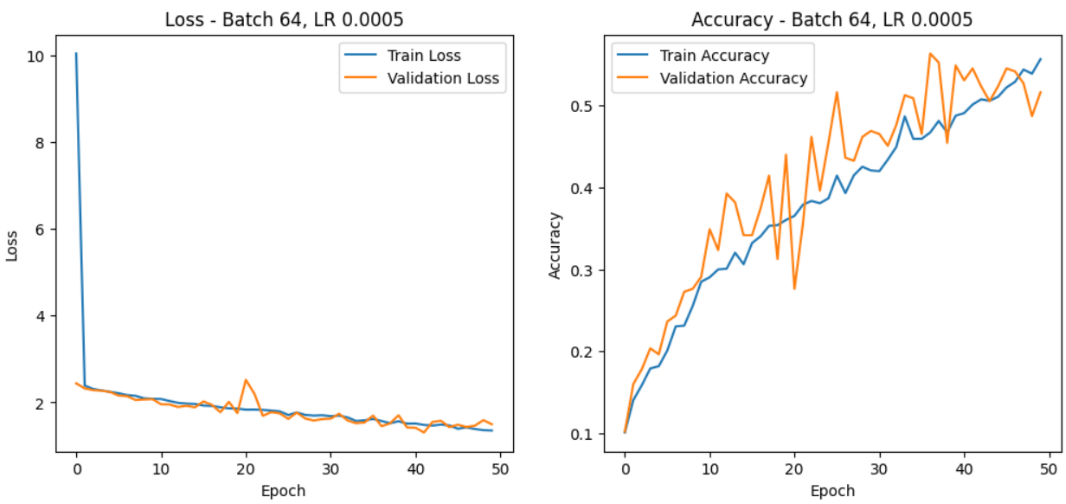
For Basic CNN Model, the values obtained at the end of 50 epochs with 32 batch size and 0.0005 learning rate are: Train Loss: 1.4427, Train Acc: 0.5241, Val Loss: 1.2406, Val Acc: 0.6218 .



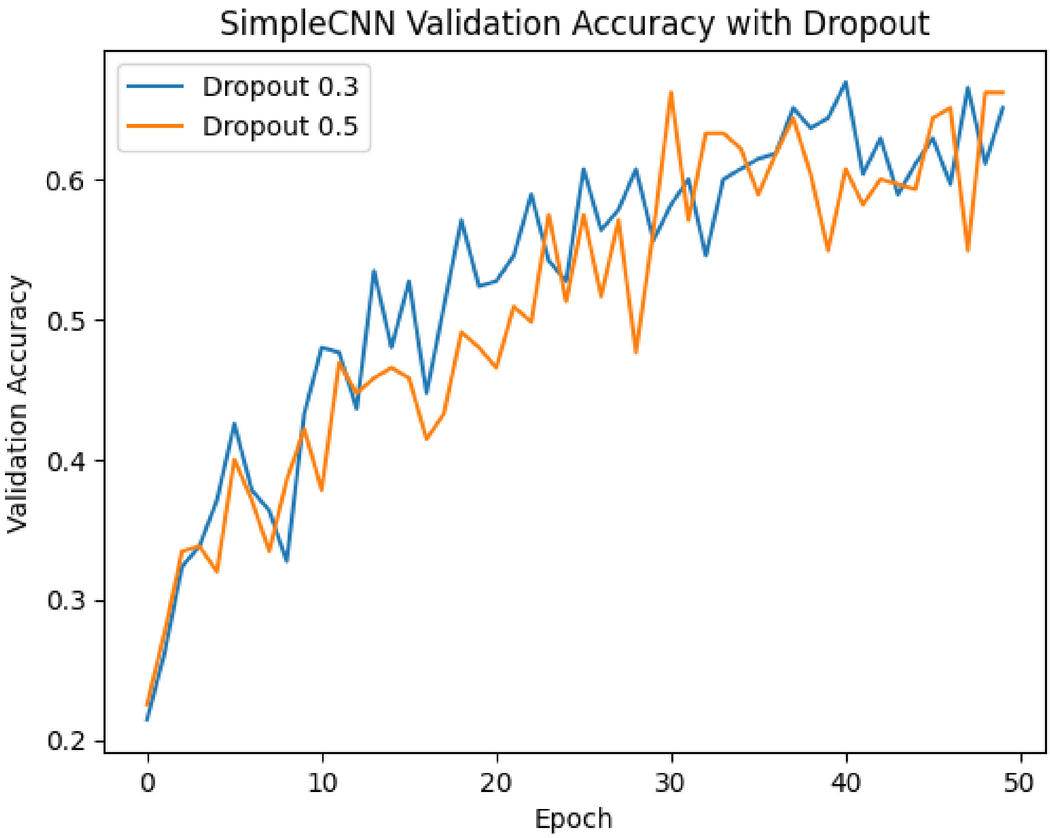
For Basic CNN Model, the values obtained at the end of 50 epochs with 64 batch size and 0.001 learning rate are: Train Loss: 1.6477, Train Acc: 0.4423, Val Loss: 1.7629, Val Acc: 0.4509



For Basic CNN Model, the values obtained at the end of 50 epochs with 64 batch size and 0.0001 learning rate are: Train Loss: 1.0207, Train Acc: 0.6659, Val Loss: 1.1157, Val Acc: 0.6327



For Basic CNN Model, the values obtained at the end of 50 epochs with 64 batch size and 0.0005 learning rate are: Train Loss: 1.5096, Train Acc: 0.4877, Val Loss: 1.4219, Val Acc: 0.5491

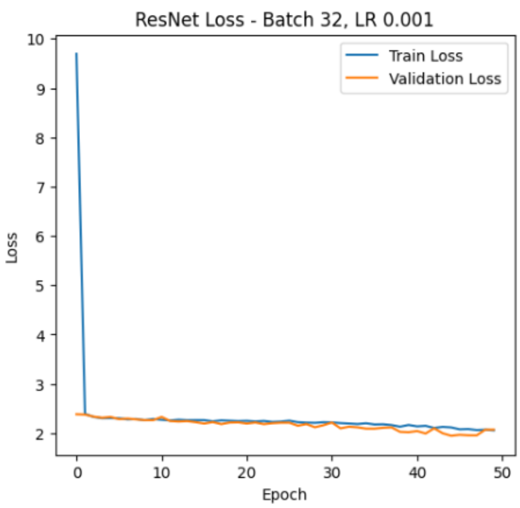


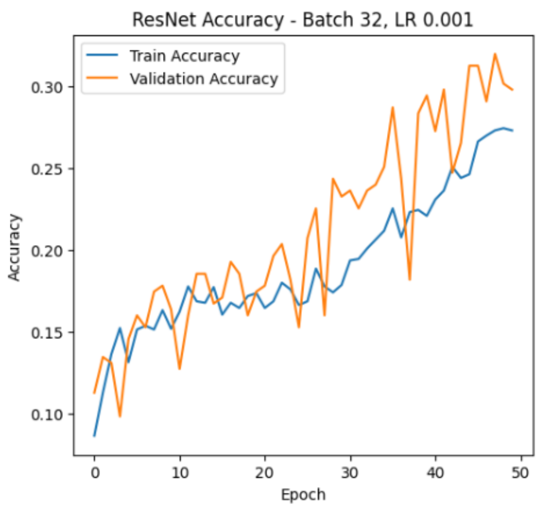
For Basic CNN, the values obtained at the end of 50 epochs with 32 batch size, 0.0001 learning rate and 0.3 dropout value are: Val Acc: 0.6509.

For Basic CNN, the values obtained at the end of 50 epochs with 32 batch size, 0.0001 learning rate and 0.5 dropout value are: Val Acc: 0.6618.

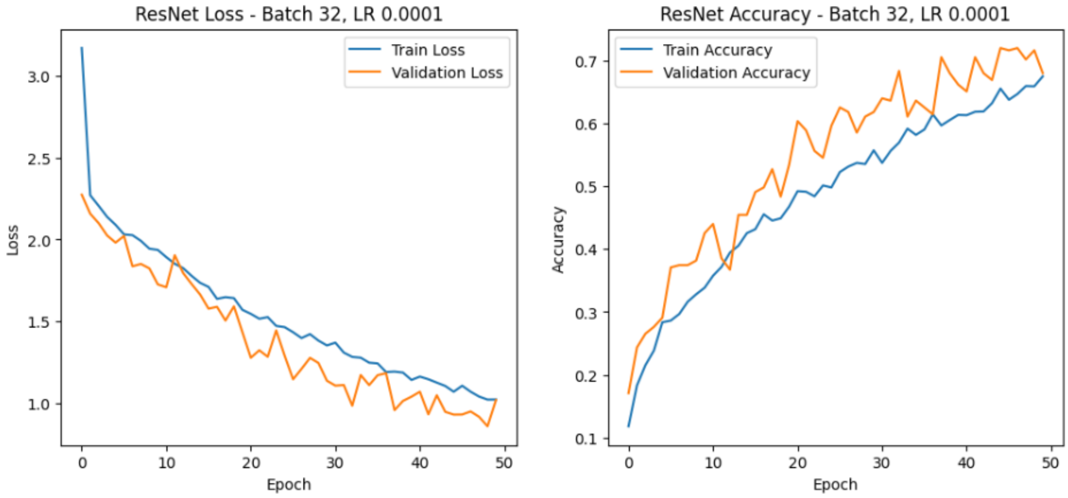
The best validation accuracy was achieved with a dropout rate of 0.5, a batch size of 64, and a learning rate of 0.0001. This configuration resulted in a validation accuracy of 0.6618, and the corresponding test accuracy was measured as 0.6271. These results indicate that applying dropout also improved generalization in the simpler architecture.

Residual CNN

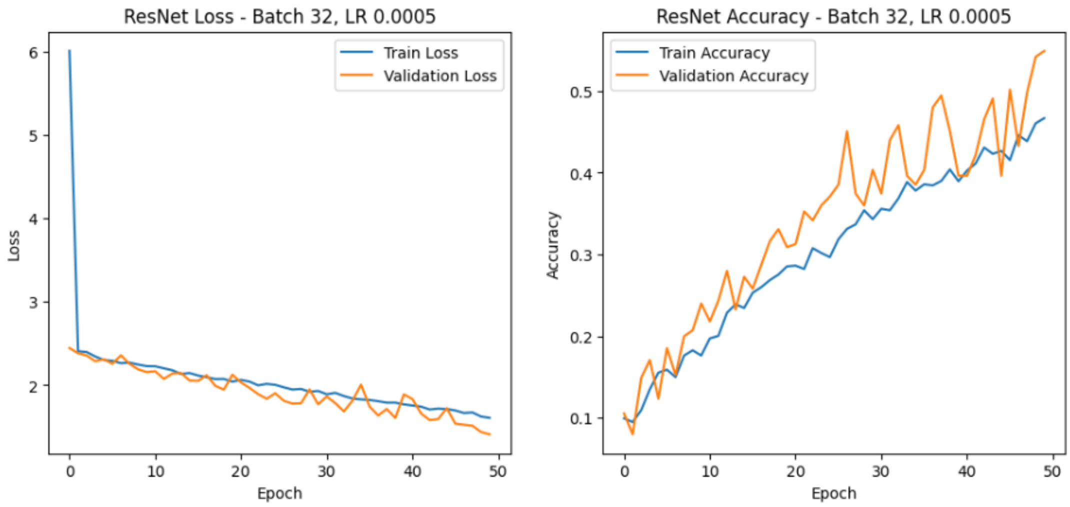




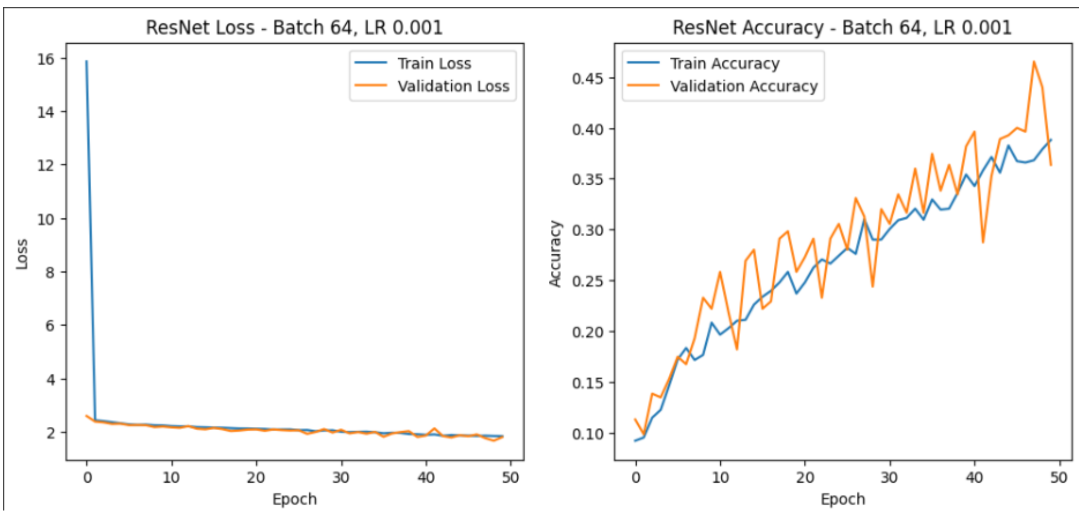
For Residual CNN Model, the values obtained at the end of 50 epochs with 32 batch size and 0.001 learning rate are: Train Loss: 1.7279, Train Acc: 0.4277, Val Loss: 1.5083, Val Acc: 0.5236



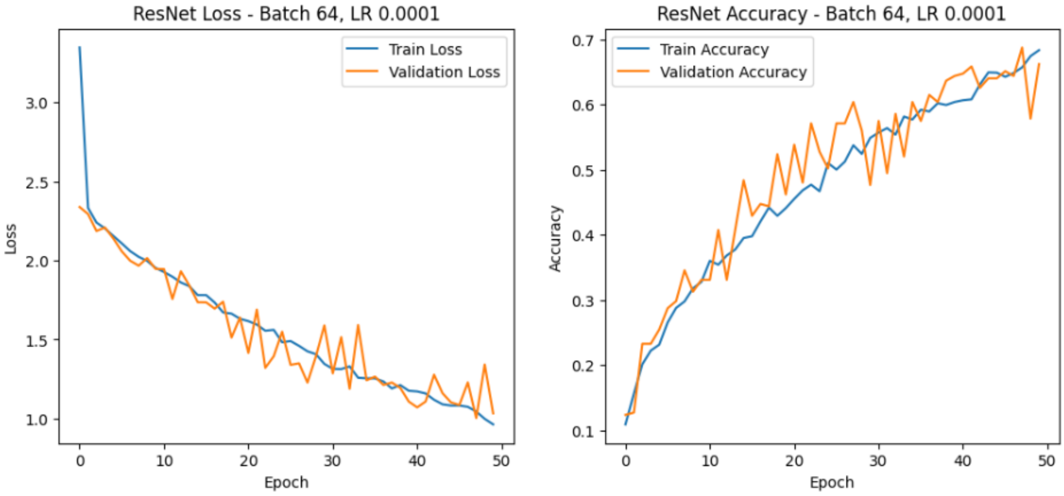
For Residual CNN Model, the values obtained at the end of 50 epochs with 32 batch size and 0.0001 learning rate are: Train Loss: 0.9877, Train Acc: 0.6718, Val Loss: 0.8029, Val Acc: 0.7273 These are the values with the best validation accuracy among the ResNet CNN models.



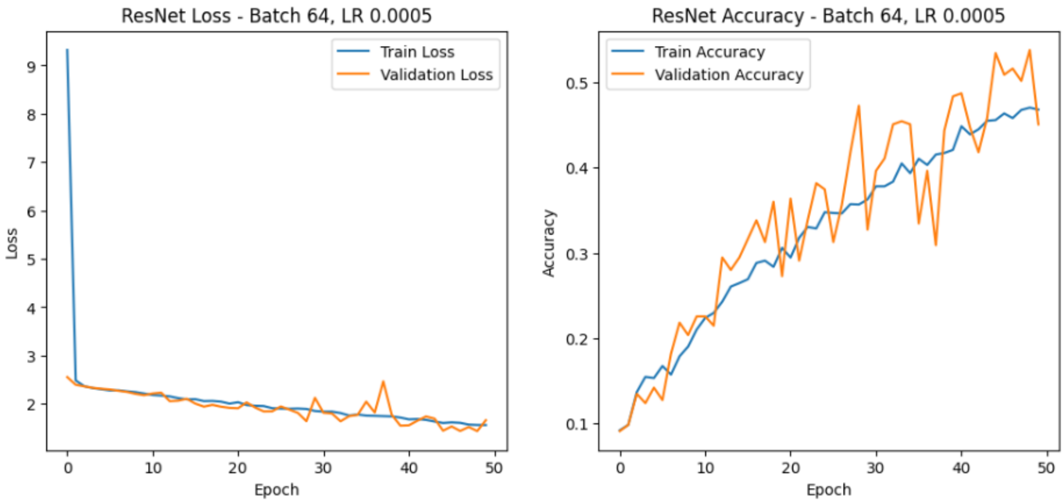
For Residual CNN Model, the values obtained at the end of 50 epochs with 32 batch size and 0.0005 learning rate are: Train Loss: 1.4256, Train Acc: 0.5155, Val Loss: 1.2580, Val Acc: 0.5673



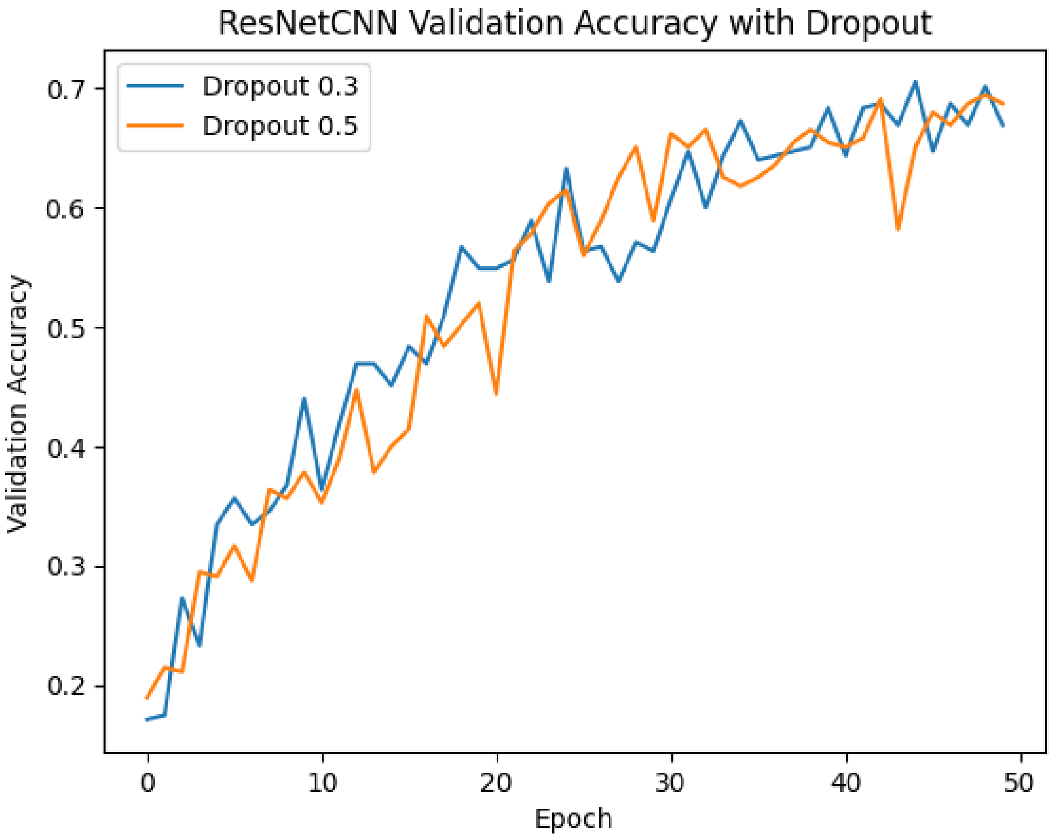
For Residual CNN Model, the values obtained at the end of 50 epochs with 64 batch size and 0.001 learning rate are:Train Loss: 1.7909, Train Acc: 0.3845, Val Loss: 1.5694, Val Acc: 0.4764 .



For Residual CNN Model, the values obtained at the end of 50 epochs with 64 batch size and 0.0001 learning rate are: Train Loss: 0.9703, Train Acc: 0.6795, Val Loss: 1.1060, Val Acc: 0.6436



For Residual CNN Model, the values obtained at the end of 50 epochs with 64 batch size and 0.0005 learning rate are: Train Loss: 1.5104, Train Acc: 0.4977, Val Loss: 1.3112, Val Acc: 0.5527



For Residual CNN, the values obtained at the end of 50 epochs with 32 batch size, 0.0001 learning rate and 0.3 dropout value are:Val Acc: 0.6836

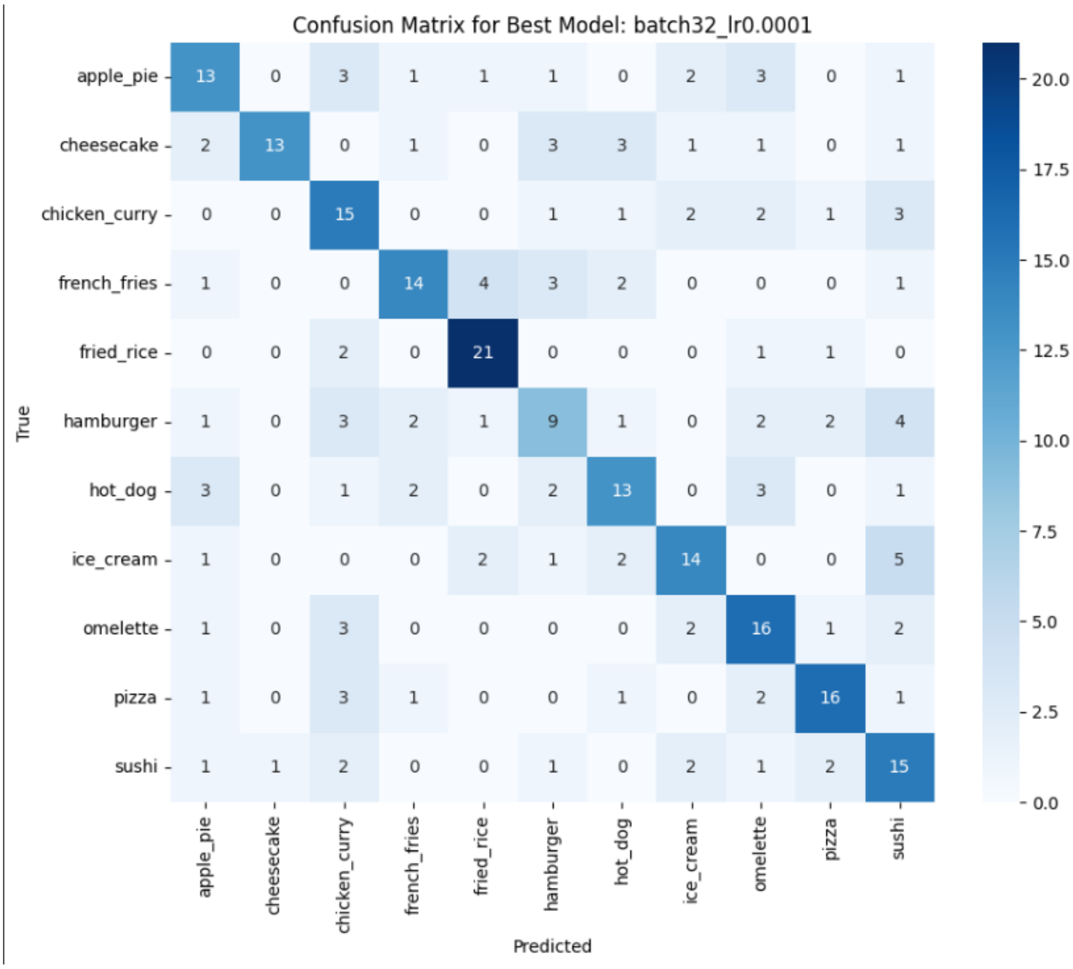
For Residual CNN, the values obtained at the end of 50 epochs with 32 batch size, 0.0001 learning rate and 0.5 dropout value are: Val Acc: 0.6873

As a result, it was determined that the Residual CNN model with a dropout rate of 0.5, a batch size of 32, and a learning rate of 0.0001 achieved the best validation accuracy of 0.6873. When this model was evaluated on the test set, it reached a test accuracy of 0.6563, showing improved generalization compared to the baseline model without dropout.

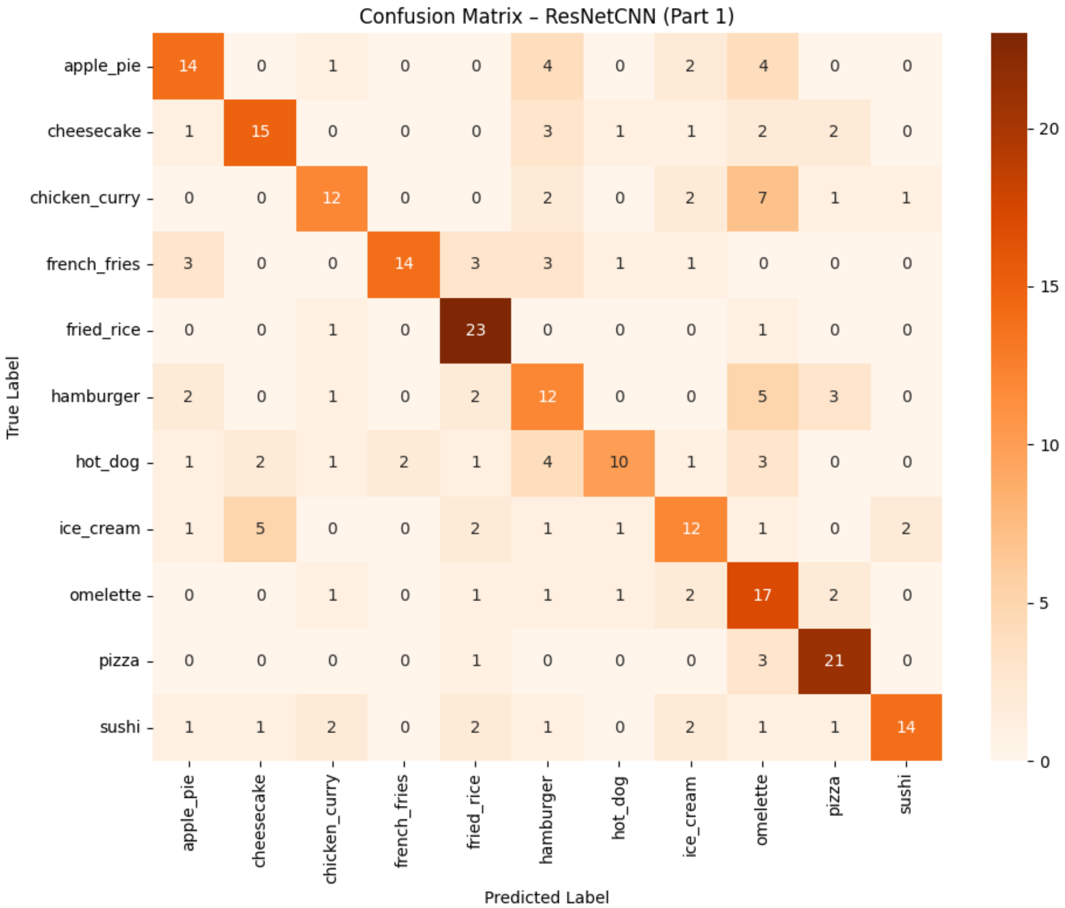
Confusion Matrix

The confusion matrix for the best scratch-trained model (batch size 32, learning rate 0.0001) shows that several classes such as fried_rice, omelette, and pizza were classified correctly with high accuracy. However, confusion was noticeable between visually similar items. For example, hamburger was often mistaken for hot_dog and pizza, while apple_pie was confused with chicken_curry and omelette. These results highlight that the model performs well overall but struggles with classes that share similar textures or visual patterns.

Basic CNN Confusion Matrix:



Residual CNN Confusion Matrix:



Part 2

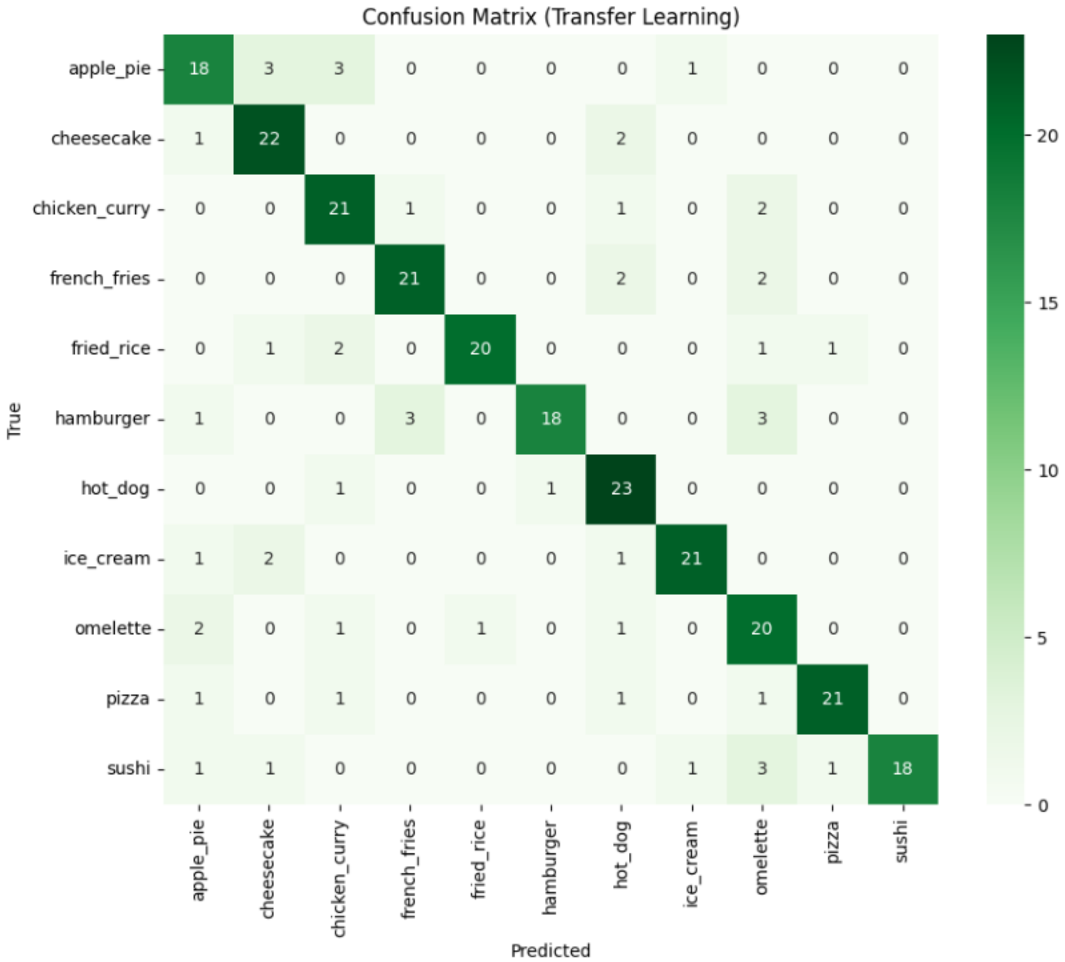
In the second part of the assignment, a transfer learning approach was applied using the MobileNetV2 architecture, which is known for its lightweight design and strong performance on image classification tasks. The model was initialized with pretrained ImageNet weights. The original classification layer was replaced with a new linear layer that outputs predictions for the 11 food classes in the Food-11 dataset.

Initially, the entire feature extractor was frozen, and only the final classifier layer was trained. This approach allowed the model to adapt the high-level features learned from ImageNet to the food classification task without modifying the lower layers. After 50 epochs of training with a learning rate of 0.0001 and a batch size of 32, the model achieved a validation accuracy of 88.36% and a test accuracy of 86.67%.

To further improve performance, a second phase of training was conducted by unfreezing the last two convolutional blocks of the network. This allowed the model to fine-tune deeper features more specific to the Food-11 dataset. The learning rate was reduced to 0.00001 to avoid disrupting the pretrained weights. After 10 additional epochs, the model reached a validation accuracy of 90.18% and a test accuracy of 88.29%, showing a clear improvement with fine-tuning.

A confusion matrix was also generated for the fine-tuned model to analyze per-class performance. Most food items were classified correctly, with minimal confusion compared to the models trained from scratch. Some errors still occurred between visually similar classes such as hamburger and hot_dog, but overall the predictions were more accurate and stable.

These results demonstrate the strength of transfer learning, especially when combined with selective fine-tuning. The model was able to generalize well with fewer training samples and significantly outperformed models trained from scratch.



Comparison: Part 1 vs Part 2

In Part 1, two custom CNN architectures—BasicCNN and ResNetCNN—were implemented and trained from scratch. Hyperparameter tuning was performed using various combinations of batch size, learning rate, and dropout rates. The best performance was obtained using dropout with a rate of 0.5 for both models. The ResNetCNN model outperformed BasicCNN, reaching a validation accuracy of 68.73

In contrast, Part 2 utilized transfer learning with MobileNetV2, a pretrained model on ImageNet. Initially, only the final classification layer was trained, and later the last two convolutional blocks were fine-tuned for further adaptation. The model reached a validation accuracy of 88.36

Confusion matrices revealed that while the scratch-trained models often misclassified visually similar categories (e.g., hamburger vs. hot dog), the pretrained model handled such cases more effectively. Predictions were more stable and accurate in the transfer learning approach.

Overall, the results clearly demonstrate that transfer learning significantly outperforms training from scratch, both in accuracy and training efficiency. It is especially advantageous when dealing with smaller datasets and visually complex categories.

Conclusion

This project compared training CNN models from scratch with using transfer learning on the Food-11 dataset. Two custom architectures (BasicCNN and ResNetCNN) were implemented and evaluated with various training configurations. ResNetCNN, especially with dropout, performed better than the basic model, achieving a maximum validation accuracy of 68.73

In Part 2, transfer learning with MobileNetV2 produced significantly better results. Training only the final layer already reached 88.36

Overall, while training models from scratch offers full control over architecture and learning, it requires more data and careful regularization. Transfer learning proved to be a much more efficient and accurate approach in this case, particularly when the dataset is limited and visually diverse. The experiments also showed that dropout can be beneficial in scratch training, but is less crucial when using pretrained networks.